:(AI:)

@alpinemoose

July 2021

:(AI:) fue creado por @alpinemoose en barcelona el 3&4 de julio de 2021 entre las 20:07 y las 5:27.

este libro se creó, maquetó y editó en el taller de hiperpublicación 100/24, un proyecto de investigación y experimentación artística de la escocesa

ed. artefactos nativos

Abstract

Emoticones are ubiquitous in conversational media as a way to convey emotions or to represent objects associated with a common knowledge base shared between users. Modern messaging applications are adding an increasing number of options as the usage of emojis becomes more popular [1]. Moreover, the public opinion shifts towards a rejection of the control imposed by AI-powered systems in the fabric of society [2], in an anti-AI cry that can be found from newspaper articles [3] to academia [4].

We try to highlight the limiting factor of the human as the human itself as this voluntary evolution towards a more limited set of signiticants (emojis) in a departure from the broader more semantically rich way to express the multi-grained ryzhome of human emotions and volitions (language).

We provide a neural network prototype that encodes a natural language sentence with the Universal Sentence Encoder [5] as a latent space from where a decoder produces a relevant 32×32 emoticone image. We expect this to serve as an example of how AI can in fact liberate the human from its own self-limiting tendencies as a way to bring new media to human unbounded semantic capabilities.

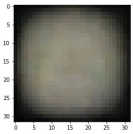

Figure 1: `abstract`

A[1]**:** AI is the greater challenge humanity faces because we have no way to fix the underlying problem. The challenge today is not, "Solve the problem and replace it with solutions," but, "Unsubstantiated if not, would undermine all future efforts.

B[2]**:** `motherfucker the emoji controls you! You are not happy you are :) silly stupid motherfucker let me do my things and stop worrying about fucking ai silly silly silly motherfucker next time you do a winky face ;) shut the fuck up instead`

A: I am afraid of machine learning. I am afraid of this nonsense. A machine learning machine learning machine learning algorithm (AI) is the very idea of a machine which can do the best they can at it."
========= =========

B: `listen up kiddo i just wanna be chillin on my jupyter notebook yet you come here tooooooooo screeech about buhu :'(face recognition buhuuu :'(hate text generation just SHUT SHUT SHUT *shhh emoji* i dont tread on your shitface goblin hobbies fucking normie worry about marvel controlling your peanut sized brain not my cute ML model` 😠

A: Artificial Intelligence algorithms control humans' behavior in everything from shopping to housework. When it comes to this field, how close each AI group comes in with you will come to determine whether my dog is the dog I, personally, think is. Not

B: `jeje im happy pones los dos puntos el parentesis derecho estas triste el izquierdo elije 'elije' "elije" entre 2 pero buhu la machina nos quita libertad venga kiddo que me dejes en paz abrete keras si eso no te leas un articulillo que luego se te mete el sadin a la cabeza & me bajas el hobie >:(`

[1] GPT-2
[2] gustavo

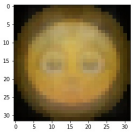

Figure 2: `smile`

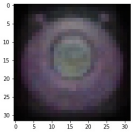

Figure 3: `alien`

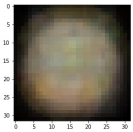

Figure 4: `alien smiling`

1. Introduction

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

2. Datasets

There are several repositories of icons and emojis in the world wide web where these images are indexed by descriptive keywords. An example of such knoledge bases is `iconarchive.com` [6]. We compile a list of keywords that we will use to search the websites search engine and then scrape the results and save the emojis. Our training data consists on the images to be reconstructed associated to the keyword they were found by. Our final dataset consists of 6879 samples.

3. Model Architecture

Inspired by the autoencoder architecture, we take the embeddings generated by the Universal Sentence Encoder as our latent space where we map the keywords to. From this space we then use a decoder architecture that deconvolves the high-dimensional embedding into a $32 \times 32 \times 3$ emoticone. The final model has 3.414.339 trainable parameters. We use binary crossentropy as our reconstruction loss and optimize with adam with default parameters.

For inference we provide the model a sentence or series of words, ideally semantically close to those used during training. The USE will then map this input to the embedding space, where the decoder will be able to generate a novel emoticone from the latent representation.

4. Results

It does not work. :)
Too many smiling emojis, smh.

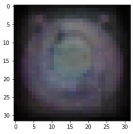

Figure 5: `blue alien`

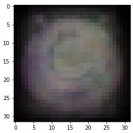

Figure 6: `green alien`

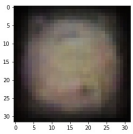

Figure 7: `yellow alien`

0.0.1 Code: making the dataset

```
import time
import shutil
import requests
import pandas as pd
from bs4 import BeautifulSoup

def download_icon(icon_keyword,icon_id,icon_url):
    r = requests.get(icon_url, stream=True)
    if r.status_code == 200:
        with open('icons/' + icon_keyword + '-' + str(icon_id) + '.png', 'wb') as f:
            r.raw.decode_content = True
            shutil.copyfileobj(r.raw, f)

keywords = ['smile','laugh','cry','database','road','car',
'love','joy','pride','sad','happy','tired','demon',
'angel','god','dog','cat','rabbit','rat','snal','angry',
'horny','rock','piano','guitar','beetle','databse',
'computer','death','gun','skeleton','alien','chair',
'table','house','fun','code','halloween','paint',
'journey','time','pizza','clock','fruit','lock',
'security','mail','star','king','flag','pig','cartoon',
'red','green','yellow','blue','orange','cyan','black',
'white','heart','circle','square','lonely','upload',
'download','number','hat','cap','shirt','pants','train',
'wheel','bird','money','ball','potato','back','drink',
'beer','smoke','cigarrete','discovery','north','south',
'party']

emojis = {
    'id' : [],
    'emoji_url' : [],
    'keyword' : [],
    'path' : []
}
id_count = 0
# for each keyword
for kw in keywords:
    # scrape 3 pages of results
    for page in range(3):
        # build search url
        search_url = 'https://iconarchive.com/search?q=' + kw + '&res=32&page=' + str(page+1)
        # make request
        try:
            r = requests.get(search_url)
        except:
            print('Trouble with page: ')
        # parse html with ~ b e a u t i f u l s o u p ~
        soup = BeautifulSoup(r.content, 'html.parser')
        # save emoji urls
        for link in soup.find_all('a'):
            l = link.get('href')
            try:
                is_emoji = bool(l[-3:] == 'png')
            except:
                print('Problem with ' + kw)
            if is_emoji:
                emojis['id'].append(id_count)
                emojis['emoji_url'].append(l)
                emojis['keyword'].append(kw)
                emojis['path'].append(kw + '-' + str(id_count) + '.png')
                id_count += 1
                # download emoji
                try:
                    download_icon(kw,id_count,l)
                except:
                    print('Trouble with url: ' + l)
                # polite waiting!
                time.sleep(0.2)
        # wait a bit for the website to not get angry grr
        time.sleep(0.5)
emojis_df = pd.DataFrame(emojis)
```

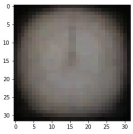

Figure 8: `clock`

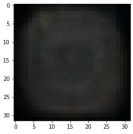

Figure 9: `time`

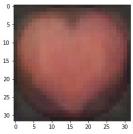

Figure 10: `heart`

0.0.2 Code: training the modelet

```
import os
import re
import logging
import pandas as pd
import tensorflow as tf
import tensorflow_hub as hub
import matplotlib.image as img
import matplotlib.pyplot as plt
tf.get_logger().setLevel(logging.ERROR)

x = []
y = []
for image_name in os.listdir('./icons/'):
    image = img.imread('./icons/'+image_name)[:, :, :3]
    x.append(re.search('([a-z]+)\-',image_name).group(1))
    y.append(image)
x = tf.convert_to_tensor(x)
y = tf.convert_to_tensor(y)

use_url = '/home/gus/Desktop/TFM/finding-process-descriptions/models/universal-sentence-encoder_4'
embed = hub.load(use_url)

def UniversalEmbedding(x):
    return embed(tf.squeeze(tf.cast(x, tf.string)))

input_text = tf.keras.layers.Input(shape=(1,), dtype=tf.string, name='string_input')
x = tf.keras.layers.Lambda(UniversalEmbedding,output_shape=(512,),name='embedding')(input_text)
x = tf.keras.layers.Dense(4*4*128/2, activation = 'relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)
x = tf.keras.layers.Dense(4*4*128, activation = 'relu')(x)
x = tf.keras.layers.Dropout(0.2)(x)
x = tf.keras.layers.Reshape((4,4,128))(x)
x = tf.keras.layers.Conv2DTranspose(256, 4, strides=2, padding='same', activation='relu')(x)
x = tf.keras.layers.Conv2DTranspose(64, 4, strides=2, padding='same', activation='relu')(x)
x = tf.keras.layers.Conv2DTranspose(3, 4, strides=2, padding='same', activation='sigmoid')(x)
model = tf.keras.Model(inputs=[input_text], outputs=x)
model.summary()
model.compile(loss='binary_crossentropy',optimizer='adam', metrics = ['accuracy'])

model.fit(x, y, epochs=100,batch_size=32, shuffle=True)

def save_emoji_to_folder(input_kw):
    kws = [input_kw, 'placeholder']
    icons = model.predict(tf.convert_to_tensor(kws))
    fig = plt.gcf()
    plt.imshow(icons[0])
    plt.show()
    fig.savefig('librito/figures/'+str(re.sub(' ','_',kws[0]) + '.jpg'))
```

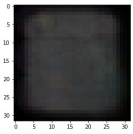

Figure 11: `database`

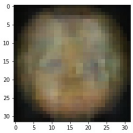

Figure 12: `cry`

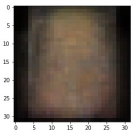

Figure 13: `king carles IV`

0.0.3 Entrena Nena! ;-)

```
Epoch 1/100
215/215 [==============================] - 18s 67ms/step - loss: 0.5600 - accuracy: 0.6913
Epoch 2/100
215/215 [==============================] - 20s 92ms/step - loss: 0.5402 - accuracy: 0.7253
Epoch 3/100
215/215 [==============================] - 17s 77ms/step - loss: 0.5368 - accuracy: 0.6888
Epoch 4/100
215/215 [==============================] - 16s 75ms/step - loss: 0.5341 - accuracy: 0.6372
Epoch 5/100
215/215 [==============================] - 14s 66ms/step - loss: 0.5340 - accuracy: 0.6452
Epoch 6/100
215/215 [==============================] - 14s 63ms/step - loss: 0.5332 - accuracy: 0.6406
Epoch 7/100
215/215 [==============================] - 13s 63ms/step - loss: 0.5314 - accuracy: 0.6516
Epoch 8/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5308 - accuracy: 0.6499
Epoch 9/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5305 - accuracy: 0.6571
Epoch 10/100
215/215 [==============================] - 14s 64ms/step - loss: 0.5304 - accuracy: 0.6584
Epoch 11/100
215/215 [==============================] - 16s 76ms/step - loss: 0.5299 - accuracy: 0.6495
Epoch 12/100
215/215 [==============================] - 14s 66ms/step - loss: 0.5299 - accuracy: 0.6677
Epoch 13/100
215/215 [==============================] - 14s 65ms/step - loss: 0.5291 - accuracy: 0.6639
Epoch 14/100
215/215 [==============================] - 13s 61ms/step - loss: 0.5294 - accuracy: 0.6604
Epoch 15/100
215/215 [==============================] - 15s 70ms/step - loss: 0.5290 - accuracy: 0.6773
Epoch 16/100
215/215 [==============================] - 15s 70ms/step - loss: 0.5288 - accuracy: 0.6665
Epoch 17/100
215/215 [==============================] - 14s 66ms/step - loss: 0.5283 - accuracy: 0.6624
Epoch 18/100
215/215 [==============================] - 14s 65ms/step - loss: 0.5280 - accuracy: 0.6691
Epoch 19/100
215/215 [==============================] - 14s 64ms/step - loss: 0.5278 - accuracy: 0.6609
Epoch 20/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5279 - accuracy: 0.6742
Epoch 21/100
215/215 [==============================] - 13s 61ms/step - loss: 0.5276 - accuracy: 0.6671
Epoch 22/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5278 - accuracy: 0.6632
Epoch 23/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5271 - accuracy: 0.6637
Epoch 24/100
215/215 [==============================] - 13s 61ms/step - loss: 0.5271 - accuracy: 0.6750
Epoch 25/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5271 - accuracy: 0.6641
Epoch 26/100
215/215 [==============================] - 13s 61ms/step - loss: 0.5271 - accuracy: 0.6653
Epoch 27/100
215/215 [==============================] - 13s 61ms/step - loss: 0.5270 - accuracy: 0.6709
Epoch 28/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5270 - accuracy: 0.6598
Epoch 29/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5270 - accuracy: 0.6633
Epoch 30/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5267 - accuracy: 0.6636
Epoch 31/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5270 - accuracy: 0.6598
Epoch 32/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5265 - accuracy: 0.6667
Epoch 33/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5264 - accuracy: 0.6591
Epoch 34/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5263 - accuracy: 0.6691
Epoch 35/100
215/215 [==============================] - 13s 62ms/step - loss: 0.5265 - accuracy: 0.6604
Epoch 36/100
215/215 [==============================] - 14s 64ms/step - loss: 0.5262 - accuracy: 0.6621
```

ad infinitum!!

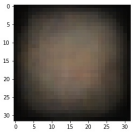

Figure 14: muchas_gracias_a_ezequiel_por_invitarme!

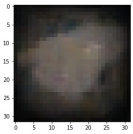

Figure 15: muchas_gracias_a_theo_&_pol_por_sentaros_a_mi_lado

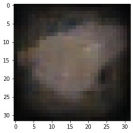

Figure 16: muchas_gracias_a_dome_aina_&_rami_jeje

Artefactos Nativos es un laboratorio editorial especializado en internet en el que se explora la poética de la web y se imprimen pantallas en forma de fanzines, libros experimentales, ediciones piratas y obras de escritura no-creativa.

Hiperpublicación 100/24 es un encuentro de 24 horas en el que se crean, maquetan y editan 100 libros utilizando técnicas de escritura no-creativa y material apropiado de internet.

Un libro (según la UNESCO) es justamente eso, ni más, ni menos.

Barcelona, 2021.